IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of

Ditlow, et al.

Serial No.: 09/943,829 Group Art Unit: 2195

Filed: August 31, 2001 Examiner: Tang, K.

For: **METHOD AND APPARATUS TO MANAGE MULTI-COMPUTER SUPPLY**

Commissioner of Patents
Alexanderia, VA 22313-1450

APPELLANTS' THIRD SUPPLEMENTAL BRIEF ON APPEAL

Sir:

In response to the Office Action mailed on November 15, 2006, Appellants again continue to attempt to appeal the rejection of claims 1-20 in the Office Action dated April 19, 2005. A Notice of Appeal was timely filed on July 19, 2005, and a Brief on Appeal was timely filed on September 19, 2005.

In an Office Action mailed on November 29, 2005, the Examiner re-opened prosecution, citing new references, US Patent Application Publication US 2002/0059625 to Kurauchi, previously-cited US Patent 6,105,053 to Kimmel et al., and newly-cited US Patent 6,016,503 to Overby et al. Appellants evaluated this new rejection and, believing that it fails to further prosecution in any meaningful manner, filed a Petition Under 37 CFR §1.181 to Re-instate Appeal on February 28, 2006, along with a Supplemental Brief to address the new rejection based on Kurauchi.

In the Office Action mailed on May 18, 2006, the Examiner again re-opened prosecution, citing a new reference, US Patent 6,601,084 to Bhaskaran et al. Appellants evaluated this new rejection and believed that it failed to meet the initial burden of a *prima*

facie rejection. A Second Supplemental Brief was submitted on August 18, 2006, in response to this Office Action, explaining how the new rejection in this Office Action failed to meet the initial burden of a prima facie rejection and was, therefore, no better than the previous two rejections currently on record.

In the latest Office Action mailed on November 15, 2006, the Examiner again re-opened prosecution. Appellants have evaluated this latest rejection and again believe that it fails to meet the initial burden of a *prima facie* rejection and that it does not serve to further advance prosecution, as explained hereinbelow.

In a telephone interview with SPE An on February 15, 2007, Appellants' representative explained that the continual re-opening of prosecution, such as is currently of record in this application, adds tremendous cost to the appeal process and does not seem to provide rejections that are any better than any of the preceding rejections of record. Accordingly, Appellants' representative requested that SPE An not allow any further re-opening of prosecution without having a significantly better rejection. A petition under 37 CFR §1.181 has been filed concurrently herewith requesting that no additional re-opening of prosecution be done in this application without signature of the director of the technical center, such signature signifying that the director has personally reviewed the new rejection and agrees with Examiner Tang and SPE An that any new rejection is significantly better than the present rejection currently of record.

Appellants believe that there is a clear difference of opinion concerning the patentability of the present application in view of the prior art currently of record and that such difference of opinion has matured to the point that the Examiner now be required to prepare an Examiner's Answer so that the prosecution can continue forward in a reasonably constructive manner.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, assignee of 100% interest of the above-referenced patent application.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative or Assignee, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-20, all of the claims presently pending in the application, stand rejected on prior art grounds.

More specifically, claims 1-4, 6-10, 12-16, and 18-20 stand rejected under 35 USC §103(a) as allegedly unpatentable over US Patent 6,601,084 to Bhaskaran et al, further in view of newly-cited US Patent 6,925,641 to Elabd, and claims 5, 11, and 17 stand rejected under 35 USC §103(a) as unpatentable over Bhaskaran/Elabd, further in view of US Patent 6,016,503 to Oberby et al.

Claims 7-16 are also understood as newly being rejected under 35 USC §101 as allegedly directed toward non-statutory subject matter.

IV. STATUS OF AMENDMENTS

An Amendment Under 37 CFR §1.116 was filed on June 20, 2005, although no claim amendments were included therein. Therefore, the version of the claims in the Appendix reflects the claim amendments of the Amendment Under 37 CFR §1.111 filed on January 4, 2005.

In the Advisory Action dated July 21, 2005, the Examiner indicated that the arguments in the Amendment Under 37 CFR §1.116 were not persuasive and that the rejections based on Robertazzi were maintained as based on the Examiner's "... broadest reasonable interpretation consistent with the specification."

A Notice of Appeal was filed on August 19, 2005, and an Appeal Brief was filed on September 19, 2005.

In an Office Action mailed on November 29, 2005, the Examiner re-opened prosecution, citing new references, US Patent Application Publication US 2002/0059625 to Kurauchi, previously-cited US Patent 6,105,053 to Kimmel et al., and newly-cited US Patent 6,016,503 to Overby et al.

A Petition Under 37 CFR §1.181 to Re-instate Appeal on February 28, 2006, along with a Supplemental Appeal Brief to address the new rejection based on Kurauchi.

In the Office Action mailed on May 18, 2006, the Examiner again re-opened prosecution, citing a new reference, US Patent 6,601,084 to Bhaskaran et al. Appellants have again evaluated this new rejection and do not believe that it furthers prosecution and have filed herewith a second Petition Under 37 CFR §1.181 to Re-instate Appeal, along with this second Supplemental Appeal Brief.

A Notification of Noncompliant Appeal Brief was mailed on September 1, 2006. and a revised version of the Appeal Brief was timely submitted in compliance thereto.

This third Supplemental Appeal Brief addresses the rejections most recently of record, as evidenced in the Office Action mailed on November 15, 2007.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Appellants' invention, as disclosed and claimed in independent claim 1, is directed to a computer-implemented method for determining a listing of hosts on a network to perform a parallel application (lines 15-16 of page 1, lines 16-19 of page 4, lines 1-3 of page 6, Figure 1), including determining a listing of all possible hosts on the network for performing the parallel application (lines 16-18 of page 1, line 19 of page 6 through line 1 of page 7). For each of the possible hosts (line 2 of page 18) a current capacity and a current utilization is determined and a difference between the current capacity and the current utilization is calculated (lines 2-4 of page 7, line12 of page 13, line 3 of page 18 through line 9 of page 19, equation [4] in Figure 7). A listing of hosts is selected from the listing of all possible hosts, based on sorting the calculated differences (lines 18-19 of page 1, lines 4-5 of page 7, lines 6-21 of page 24, Equation [6a] of Figure 7).

The remaining independent claims have similar language.

The bases in the specification for the independent claims are as follows:

Independent Claim 1

1. (Rejected) A computer-implemented method determining a listing of hosts on a network to perform a parallel application (lines 15-16 of page 1, lines 12-14 of page 2, lines 16-19 of page 4, lines 1-3 of page 6, Figure 1), said method comprising:

determining a listing of all possible hosts on said network for performing said parallel application (lines 16-18 of page 1, line 19 of page 6 through line 1 of page 7, line 18 of page 17 through line 2 of page 18, see description of "capacity subroutine" 61 at line 14 of page 21 through line 17 of page 22);

determining, for each of said possible hosts, a current capacity and a current utilization (line 4 of page 18 through line 6 of page 19, see description of "capacity subroutine" 61 at line 14 of page 21 through line 17 of page 22 and of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23);

calculating, for each of said possible hosts, a difference between said current capacity and said current utilization (lines 2-4 of page 7, line12 of page 13, line 3 of page 18 through line 9 of page 19, equation (14) on page 19, equation [4] in Figure 7, see description of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23); and

selecting from said listing of all possible hosts a listing of hosts based on sorting said calculated differences (lines 18-19 of page 1, lines 4-5 of page 7, lines 6-21 of page 24, Equation [6a] of Figure 7, see description of the "Ownership module" 92 at lines 6-9 of page 24).

Independent Claim 7

7. (Rejected) A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to

determine a listing of hosts on a network to perform a parallel application (lines 15-16 of page 1, lines 12-14 of page 2, lines 16-19 of page 4, lines 1-3 of page 6, Figure 1), said machine-readable instructions comprising:

determining a listing of all possible hosts on said network for performing said parallel application (lines 16-18 of page 1, line 19 of page 6 through line 1 of page 7, line 18 of page 17 through line 2 of page 18, see description of "capacity subroutine" 61 at line 14 of page 21 through line 17 of page 22);

determining, for each of said possible hosts, a current capacity and a current utilization (line 4 of page 18 through line 6 of page 19, see description of "capacity subroutine" 61 at line 14 of page 21 through line 17 of page 22 and of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23);

calculating, for each of said possible hosts, a difference between said current capacity and said current utilization (lines 2-4 of page 7, line12 of page 13, line 3 of page 18 through line 9 of page 19, equation (14) on page 19, equation [4] in Figure 7, see description of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23); and

selecting from said listing of all possible hosts a listing of hosts based on sorting said calculated differences (lines 18-19 of page 1, lines 4-5 of page 7, lines 6-21 of page 24, Equation [6a] of Figure 7, see description of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23).

Independent Claim 13

13. (Rejected) A computer network having a plurality of computation resources and an operating system for executing a target parallel application process using at least a subset of said plurality of computation resources, wherein said network includes a method to determine a listing of said computation resources to perform said target parallel application process (lines 15-16 of page 1, lines 12-14 of page 2, lines 16-19 of page 4, lines 1-3 of page 6, Figure 1) , said method comprising:

determining a listing of all possible said computation resources on said network for performing said parallel application (lines 16-18 of page 1, line 19 of page 6 through line 1 of page 7, line 18 of page 17 through line 2 of page 18, see description of "capacity subroutine" 61 at line 14 of page 21 through line 17 of page 22);

determining, for each of said possible computation resources, a current capacity and a current utilization (line 4 of page 18 through line 6 of page 19, see description of "capacity subroutine" 61 at line 14 of page 21 through line 17 of page 22 and of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23);

calculating, for each of said possible computation resources, a difference between said current capacity and said current utilization (lines 2-4 of page 7, line12 of page 13, line 3 of page 18 through line 9 of page 19, equation (14) on page 19, equation [4] in Figure 7, see description of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23); and

selecting from said listing of all possible computation resources a listing of computation resources based on sorting said calculated differences as said at least a subset of said plurality of computation resources to execute said target parallel application process (lines 18-19 of page 1, lines 4-5 of page 7, lines 6-21 of page 24, Equation [6a] of Figure 7, see description of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23).

Independent Claim 19

19. (Rejected) A computer network having a plurality of computation resources and an operating system for executing a target parallel application process using at least a subset of said plurality of computation resources, wherein said network includes a method to determine a listing of said computation resources to perform said target parallel application process (lines 15-16 of page 1, lines 12-14 of page 2, lines 16-19 of page 4, lines 1-3 of page 6, Figure 1), said method comprising:

means for determining a listing of all possible said computation resources on said network for performing said parallel application (equation (14) on page 19, also, lines 16-18 of page 1, line 19 of page 6 through line 1 of page 7, line 18 of page 17 through line 2 of page 18);

means for determining, for each of said possible computation resources, a current capacity and a current utilization (see description of "capacity subroutine" 61 at line 14 of page 21 through line 17 of page 22 and of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23, also, line 4 of page 18 through line 6 of page 19);

means for calculating, for each of said possible computation resources, a difference between said current capacity and said current utilization (see description of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23, also, lines 2-4 of page 7, line12 of page 13, line 3 of page 18 through line 9 of page 19, equation (14) on page 19, equation [4] in Figure 7); and

means for selecting from said listing of all possible computation resources a listing of computation resources based on sorting said calculated differences to be said at least a subset of said computation resources for executing said target parallel application process (see description of "supply subroutine" 62 at line 18 of page 22 through line 3 of page 23, also, lines 18-19 of page 1, lines 4-5 of page 7, lines 6-21 of page 24, Equation [6a] of Figure 7).

Dependent Claim 5

5. (Rejected) The method of claim 1, wherein said selecting a listing of hosts from said listing of all possible hosts further comprises a quantification of a history of each said possible host and a consideration of said history in said selecting of a listing. (lines 9-12 on page 24, "box" shown in Figure 2 that provides statistical parameters that define the host's history)

Dependent Claim 11

11. (Rejected) The signal-bearing medium of claim 7, wherein said selecting a listing of hosts from said listing of all possible hosts further comprises a quantification of a history of each said possible host and a consideration of said history in said selecting of a listing. (lines 9-12 on page 24, "box" shown in Figure 2 that provides statistical parameters that define the host's history)

Dependent Claim 17

17. (Rejected) The computer network of claim 13, wherein said selecting a listing of computation resources from said listing of all possible computation resources further comprises a quantification of a history of each said possible computation resource and a consideration of said history in said selecting of a listing. (lines 9-12 on page 24, "box" shown in Figure 2 that provides statistical parameters that define the host's history)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellants present the following grounds for rejection for review by the Board of Patent Appeals and Interferences:

1. Claims 1-20 as standing rejected under 35 U.S.C. §103(a), as based upon US Patent 6,601,084 to Bhaskaran et al. as the primary reference

2. Claims 1-4, 6-10, 12-16, and 18-20 as standing rejected under 35 U.S.C. §103(a), as based upon US Patent 6,925,641 to Elabd as a secondary reference

3. Claims 5, 11, and 17, as standing rejected under 35 U.S.C. §103(a), as based upon US Patent 6,016,503 to Overby, Jr. et al. as a second secondary reference

4. Claims 7-12, understood as standing rejected under 35 U.S.C. §101, as allegedly directed to non-statutory subject matter.

VII. ARGUMENTS

GROUND #1: THE REJECTIONS FOR CLAIMS 1-20 AS BASED ON BHASKARAN AS THE PRIMARY REFERENCE

The Examiner alleges that the present invention defined by claims 1-4, 6-10, 12-16, and 18-20 are rendered obvious when primary reference, US Patent 6,601,084 to Bhaskaran, is modified by newly-cited US Patent 6,925,641 to Elabd, and claims 5, 11, and 17 are rendered obvious when Bhaskaran/Elabd is further modified by previously-cited US Patent 6,016,503 to Oberby et al.

Appellants' Position of the Rejection Based on Primary Reference Bhaskaran

A. Bhaskaran is non-analogous art, by failing to address a parallel processing application

Appellants first point out that primary reference Bhaskaran is directed toward the problem of dynamic load balance for multiple network servers and does not address the parallel processing environment as required by the independent claims. Therefore, this reference is non-analogous art for the present invention and cannot even serve as the primary reference, since the tasks of servers on a network is not a parallel processing procedure and cannot be converted into such parallel processing without changing the principle of operation and/or purpose.

As well understood by one having ordinary skill in the art, parallel processing involves, in general, the distribution of processing tasks to be executed independently on separate CPUs. This generalized description is confirmed by the following definition from

the 6th Edition of the "Dictionary of Computer and Internet Terms", D. A. Downing, Ph.D., et al., 1998, Barron's Educational Series, Inc.:

"PARALLEL PROCESSING computation carried out at the same time on different CPUs, or on a CPU that can execute more than one instruction at the exact same time. By contrast, most multitasking is accomplished by making a single CPU switch its attention among several tasks. This is called concurrent processing or timesharing."

The description at lines 12-16 of page 2 of the specification confirms this definition: "*Multi-computer processing, hereinafter called multi-processing, involves the use of multiple computers to process a single computational task such as an application program. The term generally refers to the use of a parallel or a distributed computer. The programming of either a parallel or a distributed computer generally is referred to as "parallel programming."*

Appellants submit that the processing described in primary reference Bhaskaran is not parallel processing, as addressed by the claims of the present application and as required by the plain meaning of the claim language. Simply having the servers on a network operating in parallel or a computer directing a process to balance the loads for the network servers does not bestow the property of "parallel processing." Each computer in Bhaskaran that performs its role as a server is operating independently of the other servers and a computer executing the load balancing method is operating on its own. The servers are not involved in processing a single computational task, as would be required for there to be a parallel processing in Bhaskaran.

In contrast, parallel processing would require that more than one of the servers are concurrently involved in a common processing, such that each involved computer performs their respective assigned computing tasks for that common processing. This common processing, with each of several computers performing their respective share of the processing computations in parallel, is not present in Bhaskaran, since neither the server function nor the load balancing requires that the various computers be executing in parallel as a shared processing.

B. Bhaskaran teaches against the technique of the present invention

Moreover, Appellants submit that the method in primary reference Bhaskaran, because of its different purpose of balancing loads for servers on a network, also fails to satisfy the plain meaning of the claim language of the independent claims.

More specifically, the goal in Bhaskaran is that of equalizing the load among all of the servers. Therefore, Bhaskaran teaches against the method described by the plain meaning of the claim limitations, since the load balancing method is based upon first calculating the relative amount of power for each computer in the network, as expressed by equation 1 at line 35 in column 6. This relative power is the basis that each machine is performing its fair share in the network as determined by its relative power.

Because of this dependence on the relative power of each machine, there is no simple difference between current capacity and current utilization, as required to satisfy the plain meaning of the claim language: "... calculating, for each of said possible hosts, a difference between said current capacity and said current utilization ...."

C. Bhaskaran does not need to provide a listing of hosts

It is also noted that the load balancing Bhaskaran does not satisfy the final claim limitation and cannot be modified to do so, since the goal in this reference is to balance the load evenly through out the network, as explained at lines 42-47 of column 3. Thus, in Bhaskaran all servers will always be performing approximately their fair share as determined based on their relative power level within the network, and, as described at lines 32-38 of column 4, a skew detector determines when a server's performance is outside a tolerable range, and buckets are created to provide additional load to that server.

Thus, in the task being done in Bhaskaran, there is no need to select a listing of hosts, let along make selections based on a sorting, as required by the plain meaning of the claim language: "... selecting from said listing of all possible hosts a listing of hosts based on sorting said calculated differences."

D. Bhaskaran does not "sort" anything

To satisfy the final limitation (e.g., "*selecting from said listing of all possible hosts a listing of hosts based on sorting said calculated* difference"), the Examiner points to lines 66-67 of column 2.

In response, Appellants note that these two lines state: "*The buckets are dynamically assigned to the server having the lightest load as necessary.*" Appellants submit that this wording is merely a description of identifying the server having the lightest load, not a sorting.

That is, although identifying the server with the lightest load might be the initial step in sorting the current workload of all the servers, the process in Bhaskaran clearly stops at having identified the one having the lightest load. It clearly does not continue the sorting process and has no need to do so. Therefore, contrary to the Examiner's characterization, there clearly is no sorting being done in Bhaskaran.

E. Bhaskaran cannot serve as the primary reference

Appellants further submit that modifying primary reference Bhaskaran to select a listing of hosts would defeat the purpose therein of maintaining the work balance within the predetermined tolerance. Therefore, Appellants submit that it would be improper to modify Bhaskaran by either secondary reference currently of record or any other reference in the manner described by the final claim limitation, and, again, Appellants submit that the rejection currently of record fails to meet the initial burden of a *prima facie* rejection.

Hence, turning to the clear language of the claims, in view of the above discussion, in primary reference Bhaskaran, there is no teaching or suggestion of: ".... determining a listing of hosts on a network to perform a parallel application, said method comprising: determining a listing of all possible hosts on said network for performing said parallel application ... selecting from said listing of all possible hosts a listing of hosts based on sorting said calculated differences", as required by independent claim 1. The remaining independent claims also make reference to the parallel processing environment.

For this reason alone, Appellants submit that the rejection currently of record fails to meet the initial burden of a *prima facie* rejection based upon primary reference Bhaskaran and cannot be maintained and that claims 1-20 would be patentable over Bhaskaran for this reason alone.

GROUND # 2: THE REJECTIONS FOR CLAIMS 1-4, 6-10, 12-16, AND 18-20 AS BASED ON ELABD AS A SECONDARY REFERENCE

The Examiner alleges that the present invention defined by claims 1-4, 6-10, 12-16, and 18-20 are rendered obvious when Bhaskaran is modified by newly-cited Elabd. The Examiner alleges that primary reference "*Bhaskaran is silent on parallel processing for a load balancer*", and alleges that secondary reference "*Elabd teaches parallel processing for a load balancer (see Abstract, col. 4, lines 54-67 through col. 5, lines 1-8)*" and that one having ordinary skill in the art would have been motivated to modify Bhaskaran in accordance with Elabd "*... because it would enable processing of data at high frequency (see Abstract, col.4, lines 54-67 through col.5, lines 1-8).*"

Appellants' Position of the Rejection Based on Secondary Reference Elabd

A. Secondary reference Elabd is non-analogous art to primary reference Bhaskaran

Appellants first point out that secondary reference Elabd is directed to a problem related only to DSPs and only within a single computer, whereas primary reference Bhaskaran is directed to dynamically balancing the load of network servers, each independently and autonomously performing its own server duties. Appellants submit that these two problems and environments are completely unrelated and, therefore, non-analogous, so that secondary reference Elabd cannot be properly combined with primary reference Bhaskaran.

As pointed out in the above discussion for Bhaskaran, the modification of the primary reference to address this deficiency identified by the Examiner upon which secondary reference Elabd is relied upon, would defeat the purpose of Bhaskaran and change its principle of operation and would, therefore, be improper.

B. The rejection currently of record mischaracterizes the real issue of the Examiner's initial burden of a *prima facie* rejection

Appellants respectfully submit that rejection currently of record attempts to introduce a red herring and/or a straw man into the evaluation. In the rejection, the Examiner alleges that "*Bhaskaran is silent on parallel processing for a load balancer. Elabd teaches parallel processing for a load balancer ....*"

Appellants respectfully submit that, even if Elabd does teach a parallel processing processing for a load balancer, the real issue is not that of incorporating the parallel processing load balancing technique of Elabd into Bhaskaran.

Rather, the relevant issue is that the servers' tasks in Bhaskaran (e.g., the server load balancing) is inherently not a parallel processing, as explained earlier, and no amount of implementing a parallel processing to do the calculations defined in the limitation, even if these limitations were satisfied by the processing in Bhaskaran (which they are not, as explained above in Ground 1), would mysteriously convert the servers' tasks into a parallel processing procedure. As explained above, the servers in Bhaskaran are not executing a parallel processing procedure even if they are processing in parallel.

Whether or not the processing described in the load balancing technique of Bhaskaran were to be accomplished by a load balancer described in secondary reference Elabd is irrelevant, since the tasks of the servers is not likewise converted into a parallel processing application. Therefore, it does not matter whether or not the DSP-based method of secondary reference Elabd were to be somehow incorporated into Bhaskaran to do the load-balancing processing of Bhaskaran, since the servers are not executing a parallel processing procedure, as required by the plain meaning of the independent claim preamble.

The servers in Bhaskaran are still not executing a parallel processing application, even if the calculations that execute the load balancing were to be executed as a parallel processing application of some sort, if such is possible or desirable, as the Examiner seems to allege.

C. The rejection has no proper motivation to modify the primary reference

In paragraph 6 beginning on page 3 of the Office Action mailed on November 15, 2007, the Examiner characterizes in the rejection that "*Bhaskaran is silent on parallel processing for a load balancer.*"

The Examiner continues: "*Elabd teaches parallel processing for a load balancer (see Abstract, col. 4, lines 54-67 through col. 5, lines 1-8). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Bhaskaran's load balancer to include parallel processing because it would enable processing of data at high frequency (see Abstract, col. 4, lines 54-67 through col. 5, lines 108).*"

In response, Appellants submit that the specific problem being addressed in Elabd is perhaps best described at lines 44-67 of column 3 and can be summarized by lines 58-67: "*As can be appreciated, the conventional system is very limiting since switching algorithms can be time-consuming. In addition, the conventional system is designed to only handle data with very slow rates. Accordingly, there is a need for a DSP load management system (DLMS) that works simultaneously in conjunction with the CPUs, DSPs, memory, and memory management system to provide efficient loading or switching of algorithms and data tables into DSP memories.*"

Appellants submit that the Examiner's alleged benefit of modifying Bhaskaran ("*... to include parallel processing because it would enable processing of data at high frequency*") is taken out-of-context from that of Bhaskaran. Elabd is not at all suggesting that processing of <u>any</u> load balancing would be done at higher frequency if some sort of DSP technique is implemented. Indeed, there is no suggestion that Bhaskaran uses DSPs or that such DSPs, if used in Bhaskaran, are subject to the problem described at lines 44-67 of column 3 of Elabd.

That is, until the Examiner establishes that the problem of secondary reference Elabd being addressed is present in primary reference Bhaskaran, the motivation to modify Bhaskaran constitutes a *non-sequitur*.

D. The claim language would not be satisfied even if Bhaskaran were to be somehow modified by Elabd

As explained above, the server tasks in Bhaskaran is not a parallel processing application. Therefore, even if the processing described in Elabd were to be considered a parallel processing and somehow incorporated into Bhaskaran to calculate the load balancing, the basic functions being executed by the network servers is still not a parallel processing application. That is, primary reference Bhaskaran is directed toward dynamic load balancing for servers each processing its own autonomous tasks. This processing, even though it is done in parallel, is not a "parallel processing application", as that term of art is understood.

Incorporating the method of Elabd into Bhaskaran, even assuming *arguendo* that such incorporation could somehow be done, since the structure and function of Elabd is directed specifically toward DSPs (which Bhaskaran does not have), such incorporation would not convert the processing task of primary reference Bhaskaran into a processing that qualifies as being "parallel processing", since each server will still be performing its autonomous server tasks. These tasks are not converted into "parallel processing", simply by incorporating a load balancer that may or may not be considered as conducting some type of "parallel processing" for the load balancing task calculations.

In effect, the rejection currently of record attempts to overcome a fundamental deficiency in Bhaskaran, discussed in Ground #1 above, that the primary reference is not directed toward a "parallel processing" but, rather, a processing in parallel by servers each processing its own autonomous and independent tasks. Merely incorporating a load balancer into Bhaskaran that may or may not be executing its load balancing tasks in a manner that may or may not qualify as being a "parallel processing" does not convert the processing being executed by these servers into "parallel processing."

Moreover, even if the load balancing function of Bhaskaran is to be somehow converted into the DSB-oriented processing discussed in Elabd, the principle of operation of Bhasharan will clearly have to change if DSBs are somehow to be incorporated into a load balancing of network servers, which is, again, an entirely different problem from the load balancing of DSB functions of Elabd that is load balancing within a single computer

of functions having nothing to do with a network function, let alone network server functions. Such change in principle of operation is improper in an obviousness rejection.

GROUND # 3: THE REJECTIONS FOR CLAIMS 5, 11, AND 17 AS BASED ON OVERBY AS A SECOND SECONDARY REFERENCE

Relative to Overby, in the exemplary embodiment of the present invention, the utilization is based on CPU, memory, temporary files, and cache for each host machine and how these properties vary over time. Utilization is combined with capacity, normalized and used with a priority when selecting hosts that will solve a parallel problem.

In contrast, Overby's utilization applies to a shared resource, for example, the amount of shared memory used, and this utilization is used to manage the shared resource. This is an entirely different concept from that of managing resources on separate machines, let alone doing so for a parallel processing environment.

Therefore, Appellants submit that Overby adds nothing to overcome the basic deficiencies of Bhaskaran and/or Elabd.

More important, even if Overby teaches an historical technique, the technique in primary reference Bhaskaran is based entirely upon balancing the load at the present moment. It would be improper to incorporate an aspect of history into the selection process of Bhaskaran, since such historical factor would defeat this purpose of balancing the load at the present moment and/or change its principle of operation. Therefore, Overby cannot be combined with Bhaskaran.

GROUND # 4: THE REJECTION FOR CLAIMS 7-12 FOR STATUTORY SUBJECT MATTER

As best understood, claims 7-12 stand rejected as allegedly directed to non-statutory subject matter. Also, as best understood, the Examiner considers that the "signal-bearing medium" can be interpreted as referring to a "signal."

Appellants respectfully submit that the claim language itself expressly precludes any interpretation except that the medium is "... tangibly embodying a program of

machine-readable instructions…." As the Examiner points out, a signal is not normally considered by one having ordinary skill in the art as "tangibly embodying" anything.

Accordingly, Appellants submit that the claim language itself precludes such interpretation.

CONCLUSION

In view of the foregoing, Appellants submit that claims 1-20, all the claims presently pending in the application, are clearly enabled and patentably distinct from the prior art of record and in condition for allowance. Thus, the Board is respectfully requested to remove all rejections of claims 1-20.

Please charge any deficiencies and/or credit any overpayments necessary to enter this paper to Assignee's Deposit Account number 09-0456.

Respectfully submitted,

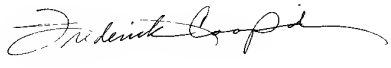
Dated: February 15, 2007

Frederick E. Cooperrider
Reg. No. 36,769

McGinn Intellectual Property Law Group, PLLC.
8231 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer Number: 21254

CERTIFICATION OF TRANSMISSION

I certify that I transmitted electronically, via EFS, this 3rd Supplemental Appeal Brief to Examiner K. Tang on February 15, 2007.



Frederick E. Cooperrider
Reg. No. 36,769

VIII. CLAIMS APPENDIX

Claims, as reflected upon entry of the Amendment Under 37 CFR §1.116 filed on June 20, 2005:

1. (Rejected) A computer-implemented method determining a listing of hosts on a network to perform a parallel application, said method comprising:

determining a listing of all possible hosts on said network for performing said parallel application;

determining, for each of said possible hosts, a current capacity and a current utilization;

calculating, for each of said possible hosts, a difference between said current capacity and said current utilization; and

selecting from said listing of all possible hosts a listing of hosts based on sorting said calculated differences.

2. (Rejected) The method of claim 1, wherein said determination of a listing of processors is itself a parallel processing application.

3. (Rejected) The method of claim 1, wherein said determination of a listing of processors is executed in real time concurrently with said parallel application.

4. (Rejected) The method of claim 1, further comprising:

providing said selected listing of hosts to an operating system controlling an execution of said parallel application.

5. (Rejected) The method of claim 1, wherein said selecting a listing of hosts from said listing of all possible hosts further comprises a quantification of a history of each said possible host and a consideration of said history in said selecting of a listing.

6. (Rejected) The method of claim 1, wherein said calculating a difference between current capacity and a current utilization further comprises:

normalizing said difference.

7. (Rejected) A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to determine a listing of hosts on a network to perform a parallel application, said machine-readable instructions comprising:

determining a listing of all possible hosts on said network for performing said parallel application;

determining, for each of said possible hosts, a current capacity and a current utilization;

calculating, for each of said possible hosts, a difference between said current capacity and said current utilization; and

selecting from said listing of all possible hosts a listing of hosts based on sorting said calculated differences.

8. (Rejected) The signal-bearing medium of claim 7, wherein said machine-readable instructions are provided to an operating system on said network such that said determination of a listing of processors is itself a parallel processing application.

9. (Rejected) The signal-bearing medium of claim 7, wherein said machine-readable instructions are provided to an operating system on said network such that said determination of a listing of processors is executed in real time concurrently with said parallel application.

10. (Rejected) The signal-bearing medium of claim 7, said machine-readable instructions further comprising:

providing said selected listing of hosts to an operating system controlling an execution of said parallel application.

11. (Rejected) The signal-bearing medium of claim 7, wherein said selecting a listing of hosts from said listing of all possible hosts further comprises a quantification of a history of each said possible host and a consideration of said history in said selecting of a listing.

12. (Rejected) The signal-bearing medium of claim 7, wherein said calculating a difference between current capacity and a current utilization further comprises:

normalizing said difference.

13. (Rejected) A computer network having a plurality of computation resources and an operating system for executing a target parallel application process using at least a subset of said plurality of computation resources, wherein said network includes a method to determine a listing of said computation resources to perform said target parallel application process, said method comprising:

determining a listing of all possible said computation resources on said network for performing said parallel application;

determining, for each of said possible computation resources, a current capacity and a current utilization;

calculating, for each of said possible computation resources, a difference between said current capacity and said current utilization; and

selecting from said listing of all possible computation resources a listing of computation resources based on sorting said calculated differences as said at least a subset of said plurality of computation resources to execute said target parallel application process.

14. (Rejected) The computer network of claim 13, wherein said method interfaces to an operating system on said network such that said determination of a listing of computation resources is itself a parallel processing application.

15. (Rejected) The computer network of claim 13, wherein said method interfaces to an operating system on said network such that said determination of a listing of computation resources is executed in real time concurrently with said parallel application.

16. (Rejected) The computer network of claim 13, said method further comprising:

providing said selected listing of computation resources to an operating system controlling an execution of said parallel application.

17. (Rejected) The computer network of claim 13, wherein said selecting a listing of computation resources from said listing of all possible computation resources further comprises a quantification of a history of each said possible computation resource and a consideration of said history in said selecting of a listing.

18. (Rejected) The computer network of claim 13, wherein said calculating a difference between current capacity and a current utilization further comprises:

normalizing said difference.

19. (Rejected) A computer network having a plurality of computation resources and an operating system for executing a target parallel application process using at least a subset of said plurality of computation resources, wherein said network includes a method to determine a listing of said computation resources to perform said target parallel application process, said method comprising:

means for determining a listing of all possible said computation resources on said network for performing said parallel application;

means for determining, for each of said possible computation resources, a current capacity and a current utilization;

means for calculating, for each of said possible computation resources, a difference between said current capacity and said current utilization; and

means for selecting from said listing of all possible computation resources a listing of computation resources based on sorting said calculated differences to be said at least a subset of said computation resources for executing said target parallel application process.

20. (Rejected) The computer network of claim 19, wherein said method interfaces to an operating system on said network such that said determination of a listing of computation resources is executed in real time concurrently with said parallel application.

IX. EVIDENCE APPENDIX

(NONE)

X. RELATED PROCEEDINGS APPENDIX

(NONE)